How to Get Live Exchange Rates in Excel using AllRatesToday API

Need live currency exchange rates inside your spreadsheet? Whether you're building a financial dashboard, tracking FX exposure, or just want today's USD/EUR rate in a cell, the **AllRatesToday API** makes it straightforward.

This guide walks you through four methods -- from a no-code Power Query approach to a fully automated VBA macro -- so you can pick the one that fits your workflow.

> **AllRatesToday updates every 60 seconds** -- most competing APIs update only hourly or daily. That means your spreadsheet always has near real-time data.

Prerequisites

Before you start, you need an AllRatesToday API key. It takes about 30 seconds:

1. Sign up for a free account at allratestoday.com.
2. Log in and go to your Profile page.
3. Copy your API key from the dashboard. It looks like `art_abc123...`

API Endpoint

Throughout this guide we'll call the same endpoint:

```
GET https://allratestoday.com/api/v1/rates?source=USD&target=EUR

Header:
  Authorization: Bearer YOUR_API_KEY
```

Replace `USD` and `EUR` with any of the 160+ supported currency codes. The response is JSON containing the exchange rate.

Method 1: Power Query (Recommended)

Power Query is built into Excel 2016+ and Microsoft 365. It handles authentication natively, refreshes on demand, and requires zero code.

1. Open Excel and go to **Data > Get Data > From Other Sources > From Web**.
2. Select **Advanced** and enter the URL: `https://allratestoday.com/api/v1/rates?source=USD&target=EUR`
3. Under **HTTP request header parameters**, add a header:
4. Name: `Authorization`
5. Value: `Bearer YOUR_API_KEY`
6. Click **OK**. Power Query will preview the JSON response.
7. Use the Power Query Editor to expand the record and select the rate field.
8. Click **Close & Load** to insert the data into your worksheet.

To refresh the rate at any time, press **Data > Refresh All** or use the keyboard shortcut `Ctrl+Alt+F5`.

> **Tip:** Power Query is the best option for dashboards and shared workbooks. The query definition is saved inside the file, and anyone who opens it can refresh -- no macros or special setup needed.

Method 2: WEBSERVICE Formula

The `WEBSERVICE` function lets you pull data from a URL directly in a formula. It's the simplest approach but has some limitations.

Basic Formula

```
=WEBSERVICE("https://allratestoday.com/api/v1/rates?source=USD&target=EUR")
```

This returns the raw JSON response as a text string in the cell.

Parsing the Rate

To extract the numeric rate from the JSON string, combine it with text functions:

```
=MID(A1, FIND("""rate"":", A1) + 7, FIND("}", A1, FIND("""rate"":", A1)) - FIND("""rate"":", A1) - 7) * 1
```

(Where `A1` contains the WEBSERVICE result.)

> **Limitations:** WEBSERVICE does not support custom HTTP headers. To use authentication, you may need to pass the API key as a query parameter if supported, or use Power Query / VBA instead. WEBSERVICE is only available on Windows desktop versions of Excel.

Method 3: VBA Macro

A VBA macro gives you full control: custom headers, JSON parsing, error handling, and even automatic refresh on a timer.

Step-by-Step

1. Press `Alt+F11` to open the VBA Editor.
2. Go to **Insert > Module** to create a new module.
3. Paste the code below into the module.
4. Close the VBA Editor and run the macro from **Developer > Macros > GetExchangeRate**.

Full VBA Code

```
Sub GetExchangeRate()
    Dim http As Object
    Dim url As String
    Dim apiKey As String
    Dim jsonResponse As String
    Dim rate As Double

    ' --- Configuration ---
    apiKey = "YOUR_API_KEY"   ' Replace with your AllRatesToday API key
    url = "https://allratestoday.com/api/v1/rates?source=USD&target=EUR"

    ' --- Make the API call ---
    Set http = CreateObject("MSXML2.XMLHTTP")
    http.Open "GET", url, False
    http.setRequestHeader "Authorization", "Bearer " & apiKey
    http.setRequestHeader "Content-Type", "application/json"
```

```
    http.Send

    ' --- Check for success ---
    If http.Status = 200 Then
        jsonResponse = http.responseText

        ' --- Parse the rate from JSON ---
        Dim rateStart As Long
        Dim rateEnd As Long
        rateStart = InStr(jsonResponse, """rate"":") + 7
        rateEnd = InStr(rateStart, jsonResponse, ",")
        If rateEnd = 0 Then rateEnd = InStr(rateStart, jsonResponse, "}")
        rate = CDbl(Trim(Mid(jsonResponse, rateStart, rateEnd - rateStart)))

        ' --- Output to cell ---
        Sheet1.Range("B2").Value = rate
        Sheet1.Range("A2").Value = "USD/EUR"
        Sheet1.Range("C2").Value = Now()  ' Timestamp
    Else
        MsgBox "API Error: " & http.Status & " - " & http.statusText
    End If

    Set http = Nothing
End Sub

' --- Optional: Auto-refresh every 60 seconds ---
Sub StartAutoRefresh()
    GetExchangeRate
    Application.OnTime Now + TimeValue("00:01:00"), "StartAutoRefresh"
End Sub

Sub StopAutoRefresh()
    On Error Resume Next
    Application.OnTime Now + TimeValue("00:01:00"), "StartAutoRefresh", , False
    On Error GoTo 0
End Sub
```

> **Tip:** Run `StartAutoRefresh` to update the rate every 60 seconds automatically. Run `StopAutoRefresh` to stop the timer. Save the workbook as `.xlsm` (macro-enabled) to keep the code.

Method 4: Google Sheets (Apps Script)

Google Sheets doesn't have WEBSERVICE, but it has something better: Apps Script with `UrlFetchApp`, which supports custom headers natively.

Step-by-Step

1. Open your Google Sheet and go to **Extensions > Apps Script**.
2. Delete the default code and paste the script below.
3. Replace `YOUR_API_KEY` with your actual key.
4. Save the script (Ctrl+S) and close the Apps Script editor.
5. Back in the sheet, use the custom function: `=getRate("USD","EUR")`

Apps Script Code

```
/**
 * Fetches the live exchange rate from AllRatesToday.
 *
 * @param {string} source  Source currency code, e.g. "USD"
 * @param {string} target  Target currency code, e.g. "EUR"
 * @return {number}        The current exchange rate
 * @customfunction
 */
function getRate(source, target) {
  var API_KEY = "YOUR_API_KEY"; // Replace with your key

  var url = "https://allratestoday.com/api/v1/rates"
          + "?source=" + encodeURIComponent(source)
          + "&target=" + encodeURIComponent(target);

  var options = {
    method: "get",
    headers: {
      Authorization: "Bearer " + API_KEY
    },
    muteHttpExceptions: true
  };

  var response = UrlFetchApp.fetch(url, options);
  var json = JSON.parse(response.getContentText());

  if (response.getResponseCode() !== 200) {
    throw new Error("API error: " + json.message);
  }

  return json.rate;
}
```

Now you can use `=getRate("USD","EUR")` in any cell. The rate updates each time the sheet recalculates or when you manually refresh.

> **Note:** Google Sheets caches custom function results. To force a refresh, add a volatile parameter: `=getRate("USD","EUR",NOW())`. This recalculates every time the sheet recalculates.

Which Method Should You Use?

Method	Auth Header	Auto-Refresh	Platform
Power Query	Yes	Manual / scheduled	Excel 2016+, M365
WEBSERVICE	No	On recalc	Windows Excel only
VBA Macro	Yes	Timer-based	Windows/Mac Excel
Apps Script	Yes	On recalc / trigger	Google Sheets

Our recommendation: Use **Power Query** for Excel dashboards and **Apps Script** for Google Sheets. Both support the Authorization header natively and are the most reliable long-term solutions.

Frequently Asked Questions

How often does AllRatesToday update exchange rates? AllRatesToday updates exchange rates **every 60 seconds**, providing near real-time data for over 160 currencies. Most competing APIs update only hourly or daily.

Is the AllRatesToday API free? Yes. AllRatesToday offers a free tier that includes access to all currency pairs and real-time rates. Sign up at allratestoday.com to get your API key.

Can I get multiple currency pairs at once? Yes. You can specify multiple target currencies by separating them with commas in the `target` parameter, e.g., `?source=USD&target=EUR,GBP,JPY`.

Does WEBSERVICE work on Mac Excel? No. The `WEBSERVICE` function is only available on Windows desktop versions of Excel. On Mac, use Power Query or VBA instead.

Can I use AllRatesToday with Google Sheets? Absolutely. Use Google Apps Script with the `UrlFetchApp` service to call the AllRatesToday API and return live rates directly into your spreadsheet cells as shown in Method 4 above.